sucimaPix

A software suite for pixel detector data analysis

Antonio Bulgheroni (INFN – Roma III)

EUDET – JRA1 Meeting – Geneva – 4th April 2006

Contents

- A series of questions: what's that sucimaPix? When will it be ready? What do I need to run it? Where can I read more? How difficult is to analyse my own pixel detector? How does it work? What's next?
- An example...
- Basic idea and working principle
- Code structure

What's that?

- sucimaPix is data analysis software framework providing all the general and standard tools required by pixel detector characterization
- It's C++ coded and exploits the advantages of Object Orientation
- It's fully embedded in Root: can be used as a standalone program or runned interactively from a Root shell.

When will it be ready?

- It’s available right now! Just download, compile and enjoy it!
- It was born 2 years ago within the SUCIMA collaboration (mainly developed and used in Como) to analyse data taken with M5, MimoRoma, Successor5, MimoTera and now also on Mimo* 2
- It was the first attempt to move from the standard Paw/Fortran enviroment toward the new (and future) Root/C++ one
- Since then, it evolved very much and pretty quickly becoming a stable and unique software tools in our labs.

What do I need to run it?

- You need a personal computer with Root (> 3.5) installed.
- sucimaPix can be compiled under any operating system where Root is compilable (Linux, Windows, Mac Os). It has been successfully ported under Windows in one night only!
- In pricinple you can download and use binaries but compiling the source is reccomended.
- The development and the stable CVS snapshots can be downloaded from our server (address at the end).
- ... And of course you need a sample of data to analyze

Where can I read more?

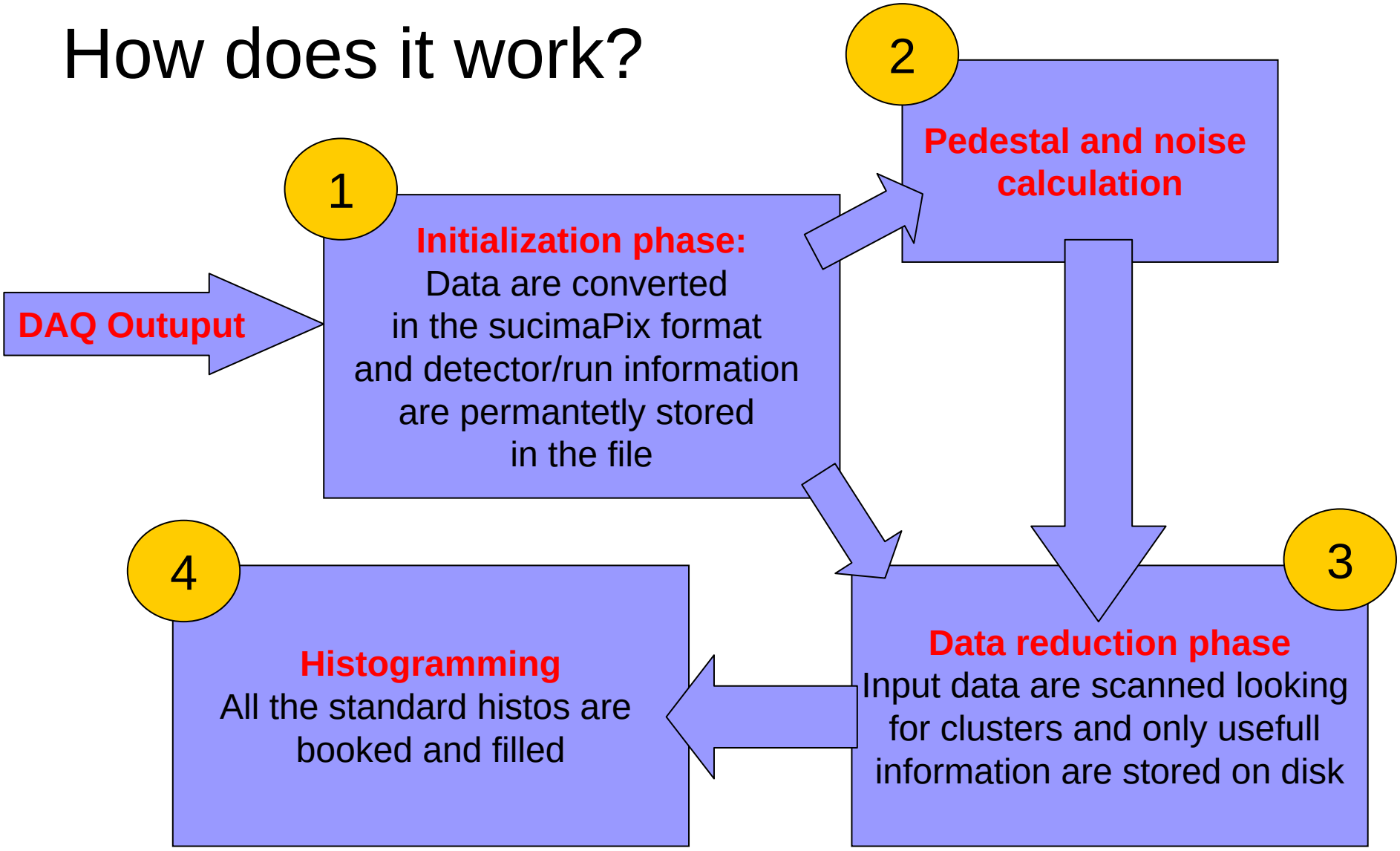
- sucimaPix is well documented and the documentation is available on the web: http://sucimaweb.dipscfm.uninsubria.it/doc/sucimapix
- The documentation is done using the Root style (THtml) and includes also Root classes.
- You are invited to report bugs to our BugZilla http://sucimaweb.dipscfm.uninsubria.it/bugzilla
- You can subscribe to the developers mailing list sucimapix-dev@sucimaweb.dipscfm.uninsubria.it to stay tuned with the latest release.

How difficult is to run it with my sensor?

- If your DAQ output data format is already compatible with the sucimaPix one (the LEPSI DAQ is supported) then it's trivial: just fill in the configuration file
- sucimaPix is a completely general enviroment. The source has been written in a dynamic way, so there are no limitation in the size and number of pixels in your detector.
- If your DAQ output is not supported, a new 'converter' has to be developed.
- All the "standard" histograms are provided but, if you know a little of Root and some basics of C++, you can easily add histos and personalize them.

A real example with real data...

How does it work?



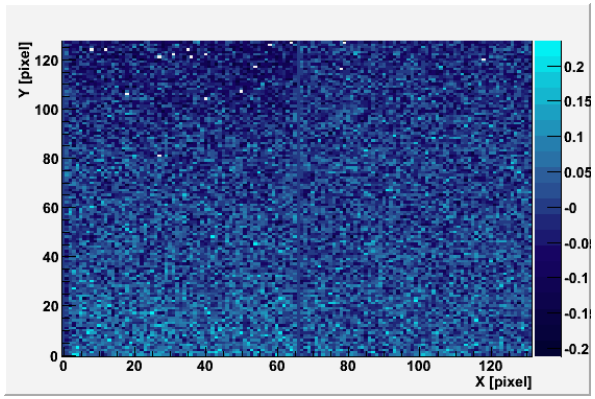
Initialization phase 1

- Data coming from the DAQ are converted in a TTree of TPixelMatrix.
- Also detector and run information are saved in the file
- This is the slowest part of the software execution and may seem to be useless if only one kind of detector and one DAQ is used. Moreover it is compressing data! Real case: 3.2 GB of Mimo*2 data compressed to 768 MB!
- It is providing a common entry point for all the rest of the analysis that can proceed in the same way regardless the specific detector and DAQ specifications.

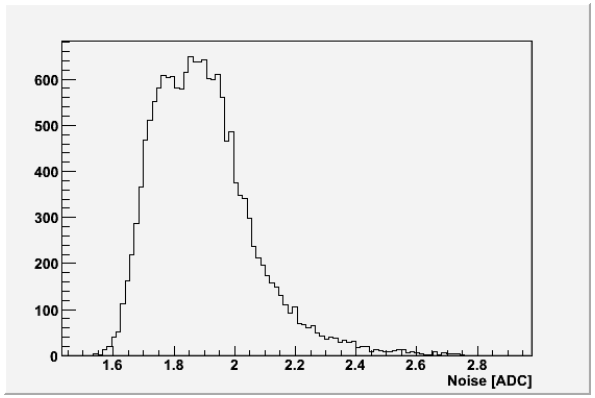
Pedestal and noise calculation 2

- For each pixel both the pedestal and noise are calculated from a run without source. For the future, also automatic pedestal estimation and tracking are foreseen.
- The noise of each pixel is crucial because all the following “cuts” are given in SNR units.

Pedestal map

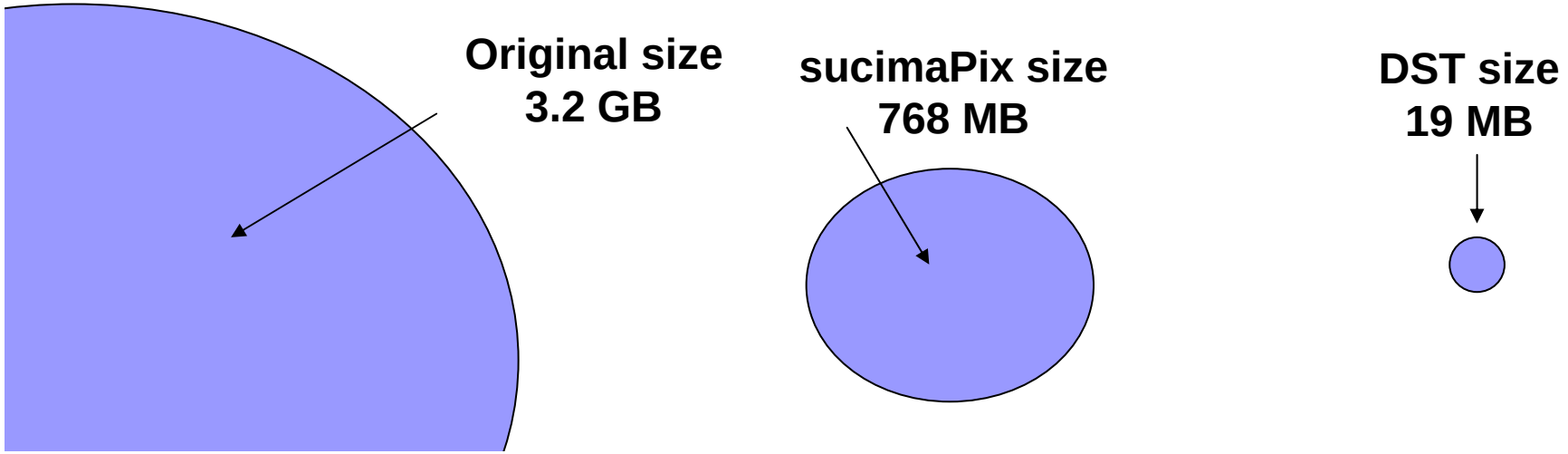


Noise distribution

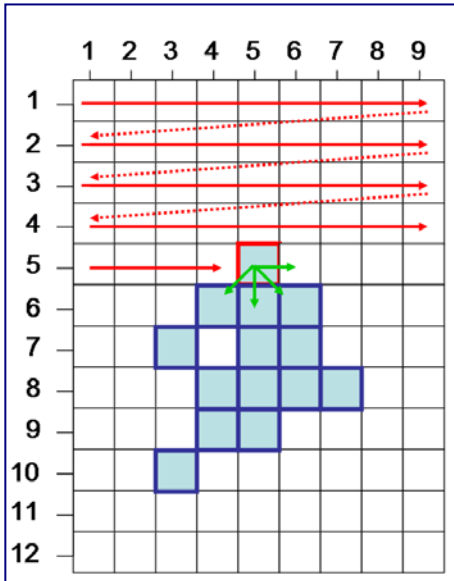


Data reduction and DST production

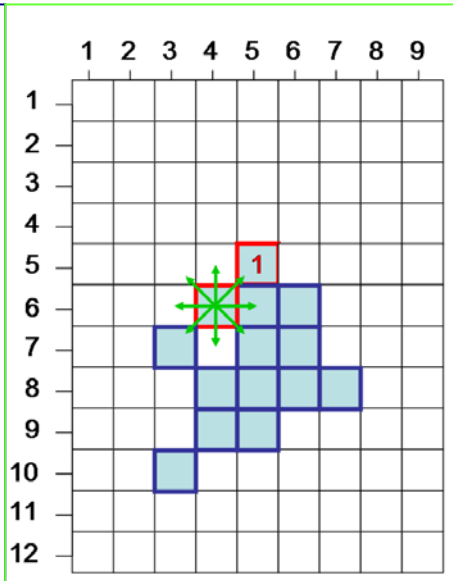
- Run data are pedestal sub'ed and common mode corrected.
- The frame is scanned for clusters. Two different algorithms are available: standard fixed frame (n x m pixels) and a recursive one.
- The recursive algorithm is faster and more general...



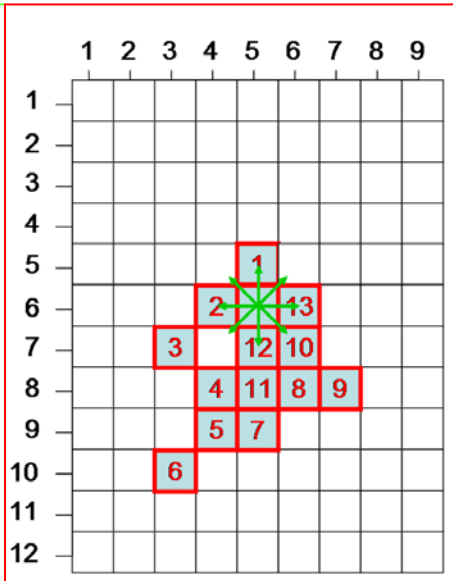
Recursive cluster search



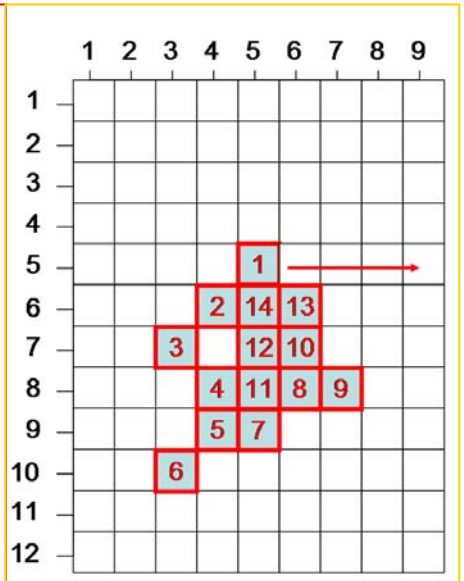
Matrix scanned until a first pixel above threshold is found. The first 4 copies of *Add&Check* are generated.



If the added pixel is found above threshold, *Add&Check* generates other 8 copies of itself for neighbouring pixels



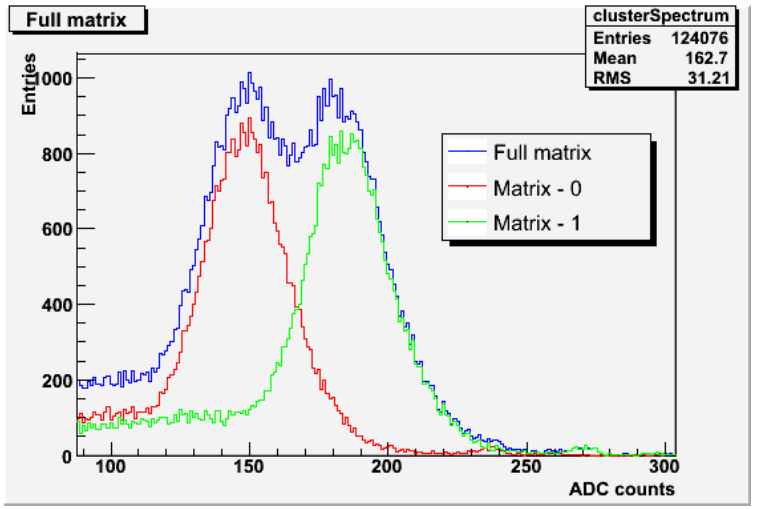
If *Add&Check* doesn't find anything else to add, it returns to a previous level until all the copies exit.

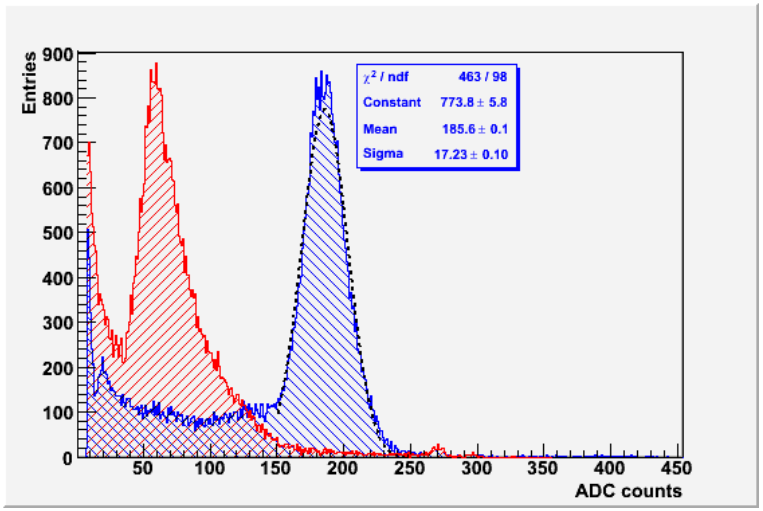


The matrix scanning can restart from where it was stopped.

Histogramming 4

- With the information stored in the DST, producing histos is very simple and extremely quick. In this way you can easily compare the results of different cuts.
- This last phase can be easily personalized in order to build your favourite histos and distributions.





Source code structure

- Three shared libraries that can be also dynamically loaded into Root

libEvent.so (.dll) **libAnalib.so (.dll)** **libGUI.so (.dll)**

- A bunch of converters and one executables for each of the other analysis steps:

1 **ascii2root (.exe)**
lepsiconv (.exe)
teraconv (.exe)
....

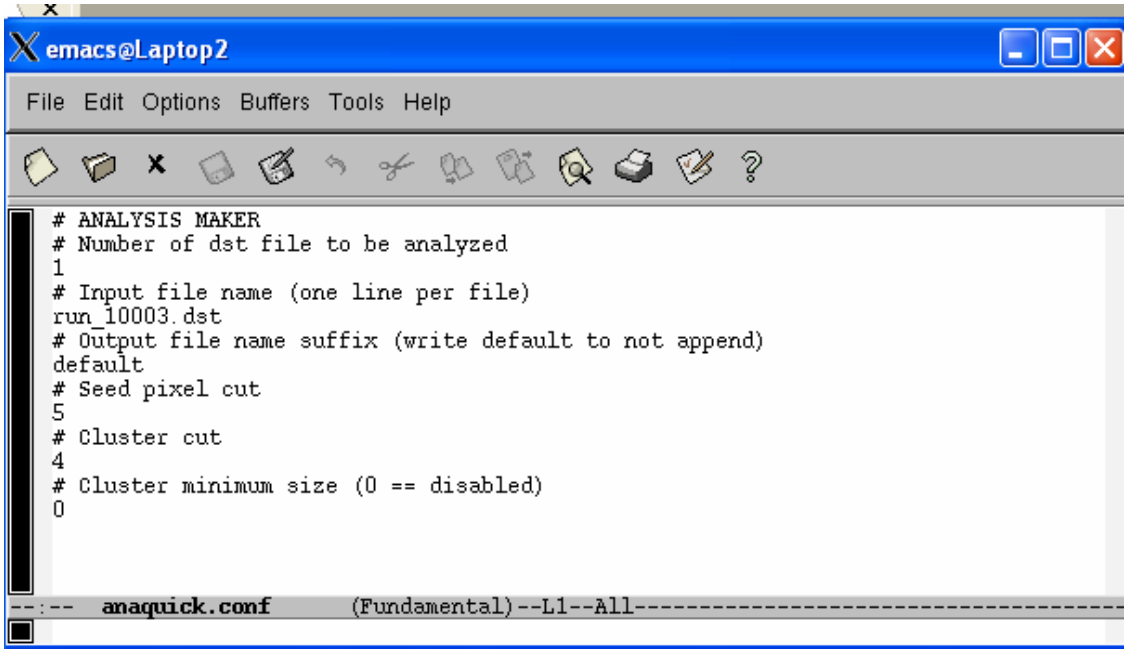
2 **pedequick**

3 **dstquick**

4 **anaquick**

How can I run it?

- A first very primitive GUI has been written but now we will replaced it with a new one Qt-based.
- Until then the only way to run it is executing the binary from a shell command line.
- The binaries are steered by a ASCII configuration file



What’s next?

- Implement the Qt-based GU interface
- Integrate the sucimaPix framework in a telescope enviroment in order to deal with many sensors making available 3D coordinates for track reconstruction.
- As an intermediate step, integration of INFN strip telescope.
- On a best effort basis, your wish list...

Anonymous CVS

CVSROOT=:pserver:anonymous@sucimalab.dipscfm.uninsubria.it/root

cvs loging (when prompted for password type sucimaPix)

cvs checkout –r v2-0-4 –P sucimapix

Latest stable release is v2-0-4.

The first “LEPSI compatible” version will be released before the end of this week and will be v2-0-5.

Send me (antonio.bulgheroni@roma3.infn.it) to join the sucimapix-dev mailing list

Conclusions

- sucimaPix, a software suite, Root-embedded has demonstrated itself to be stable and can profitable used for pixel detector characterization.
- The C++ OO code has been carefully written in order to be fully dynamic and portable.
- The documentation is produced while writing the code and available on line.
- The user and developer communities need to be enlarged.